REMARKS

The Examiner has objected to Claim 3 due to informalities. Applicant respectfully notes that such objection has been avoided in view of the amendments made to Claim 3.

The Examiner has rejected Claims 1-3, 5-8, 12, 14-19, 21-24, 28, 30-35, 37-40, 44, 46-51, 53-56, 60, 62-67, 69-72, 76, 78-83, 85-88, 92, 94-96, and 98 under 35 U.S.C. 103(a) as being unpatentable over Cozza (U.S. Patent No. 5,649,095) in view of Arnold et al. (U.S. Patent No. 5,442,699), and further in view of Pietrek ("Peering Inside the PE: A Tour of the Win 32 Portable Executable"). Applicant respectfully disagrees with such rejection.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed.Cir.1991).

With respect to the first element of the *prima facie* case of obviousness and, in particular, the obviousness of combining the aforementioned references, the Examiner has argued that "it ... would have been obvious to the ordinary person skilled in the art at the time of [the] invention to employ the teachings of Arnold in the virus scanning of Cozza by creating hashes of the data, including the resources, of the compressed file and comparing it to hashes of known viral patterns." To the contrary, applicant respectfully asserts that it would not have been obvious to combine the teachings of the Cozza and Arnold references, especially in view of the vast evidence to the contrary.

The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. In re Mills, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990). Although a prior art device "may be capable of being modified to run the way the apparatus is claimed, there must be a suggestion or motivation in the reference to do so." 916 F.2d at 682, 16 USPQ2d at 1432.).

Applicant respectfully points out that the Cozza reference teaches that "[i]f [there is a] need to scan for one or more viruses then a check is made to see whether the file is compressed" (Col. 7, lines 16-19). Cozza further discloses that "[a] compressed file may have already been decompressed" but that "[i]f the file does require decompression, then it is decompressed" (Column 7, Lines 20-24). In fact, Cozza only teaches that after such decompression, the file is scanned (see steps 84, 86, 90 and 94 in Figure 4D of Cozza).

However, the Arnold reference merely relates to "automatic[ally] search[ing] for one or more patterns which may be contained within a body of text or computer data in encrypted form and, more particularly, to searching for encrypted patterns within computer viruses" (Column 1, Lines 18-22 – emphasis added). In fact, applicant respectfully points out that Arnold expressly discloses "detect[ing] and locat[ing] patterns that are present within data that has been encrypted" (see Abstract), where "for those patterns where it is of interest to detect, in the text 44, encryptions of the pattern rather than just the pattern itself (e.g. patterns from self-encrypting computer viruses), the signatures may be subjected to an invariant transformation appropriate to the type of encryption" (Col. 4, lines 40-45). Clearly, Arnold only teaches searching for patterns within encrypted data. To this end, there would have been no motivation or suggestion to combine Arnold, which only relates to searching for patterns within encrypted data, with the teachings of Cozza, which only scans data after such data is decompressed. Thus, the Examiner's proposed combination is inappropriate.

Further, the fact that Cozza teaches that decompression takes place before scanning for viruses actually *teaches away* from "creating hashes of data... of the compressed file" in order to "scan the files as quickly as possible" (emphasis added), as alleged by the Examiner to be taught in Arnold. It is improper to combine references where the references teach away from their combination. *In re Grasselli*, 713 F.2d 731, 743, 218 USPQ 769, 779 (Fed. Cir. 1983).

Thus, applicant respectfully asserts that the first element of the *prima facie* case of obviousness has not been met, as noted above.

More importantly, applicant also respectfully asserts that the third element of the *prima facie* case of obviousness has not been met by the prior art reference excerpts relied on by the Examiner. For example, with respect to the independent claims, the Examiner has relied on Col. 2, lines 54-65 and Col. 6, lines 6-45 from the Cozza reference to make a prior art showing of applicant's claimed "reading resource data within said packed computer file, said resource data specifying program resource items used by said known computer program and readable by a computer operating system without dependence upon which unpacking algorithm is used by said packed computer file" (see this or similar, but not necessarily identical language in the independent claims).

Applicant respectfully asserts that the above reference excerpts relied on by the Examiner merely teach that "the file's cache information is checked to see if it is marked as having been previously infected by some virus which changes a file's resource fork size" (Col. 6, lines 21-23 – emphasis added) and that "[i]f a file's cache information is not marked as having been previously infected ... then the file's current resource fork size is compared with the resource fork size stored in the file's cache information... to see if they are within some predetermined tolerance" (Col. 6, lines 21-23 – emphasis added).

Further, the excerpts teach that "the resource fork... may contain a kind of small database which is used to contain many kinds of data, including application code, icons,

preferences, strings, templates, and other such items" (Col. 2, lines 56-60 – emphasis added). Also, the excerpts disclose that "[i]f the compressed file sizes… are different or if there are some viruses that could infect this file without changing its compressed size… then fork size information for this file is obtained," which "could involve decompressing the file, opening the file, or executing some special system or other code in order to obtain this information" (Col. 6, lines 14-20 – emphasis added).

However, merely disclosing comparing the resource fork size with a cached resource fork size, where the resource fork may contain a small database including application code, icons, preferences, strings, and templates, as in Cozza, fails to teach "reading resource data within said packed computer file, said resource data specifying program resource items used by said known computer program" (emphasis added), in the context claimed by applicant. Further, merely teaching that fork size information for a file may be obtained by decompressing the file, opening the file, or executing some special system, as in Cozza, does not teach "reading resource data within said packed computer file, said resource data… readable by a computer operating system without dependence upon which unpacking algorithm is used by said packed computer file" (emphasis added), in the context claimed by applicant.

Further still, the Examiner has argued that Cozza teaches that "the compressed file is not decompressed in order to read the resource forks information." Applicant respectfully disagrees and notes that Cozza teaches that "fork size information for this file is obtained" and that "[t]his could involve decompressing the file" (Col. 6, lines 17-19 – emphasis added).

Additionally, with respect to the independent claims, the Examiner has relied on "RESFORKLEN" in Figure 5 of Cozza, in addition to Col. 2, lines 54-65 in Cozza to make a prior art showing of applicant's claimed technique "wherein said generated fingerprint data includes a number of program resource items specified within said resource data of said packed computer file" (see this or similar, but not necessarily identical language in the independent claims).

Applicant respectfully asserts that "RESFORKLEN" in Figure 5 of Cozza simply relates to a file resource fork length, as shown in Figure 5, where such length is defined by an associated size in bytes. Clearly, only disclosing a length of a file resource fork in terms of bytes, as in Cozza, fails to specifically meet applicant's claimed technique "wherein said generated fingerprint data includes a number of program resource items specified within said resource data of said packed computer file" (emphasis added), as claimed.

In addition, Col. 2, lines 54-65 in Cozza simply discloses that "one fork of a file…may contain a kind of small database which is used to contain many kinds of data, including application code, icons, preferences, strings, templates, and other such items." However, simply disclosing that a file may contain many kinds of data, as in Cozza, fails to suggest a technique "wherein said generated fingerprint data includes a number of program resource items specified within said resource data of said packed computer file" (emphasis added), as claimed by applicant.

Still yet, the Examiner has argued that "[i]t would have been obvious in this combination that because the file contains the resource fork and resource items, and the hash is taken of the file, the signature includes a number of resource items specified within the resource fork."

Applicant respectfully disagrees and asserts that applicant claims that "said resource data of said packed computer file is processed to generate fingerprint data" where "said generated fingerprint data includes a number of program resource items specified within said resource data of said packed computer file," when read in context. Clearly, the fingerprint claimed by applicant is generated from processed resource data, in the context claimed, and is not merely a hash of a file that includes the resource fork and resource items, as alleged by the Examiner. Thus, it would not have been obvious for "said generated fingerprint data [to include] a number of program resource items

specified within said resource data of said packed computer file," in the context applicant specifically claims.

Further, the Examiner has relied on Figure 5 in Cozza to make a prior art showing of applicant's claimed technique "wherein said generated fingerprint data includes a flag indicating which data is included within said generated fingerprint data" (see this or similar, but not necessarily identical language in the independent claims).

Applicant respectfully asserts that the only "flag" shown in Figure 5 of Cozza is simply utilized to indicate the presence of a virus. In addition, applicant points out that the flag is included in a scan information cache file which "includes data that has been accumulated during the scanning of files" (Col. 5, lines 17-21). Clearly, a flag that indicates whether a virus is present in scanned data, as in Cozza, fails to teach that "generated fingerprint data includes a flag indicating which data is included **within said generated fingerprint data**" (emphasis added), as claimed.

Moreover, the Examiner has argued that "[i]t further would have been obvious that because the fingerprint data represented the file during comparison, and the flags of Cozza indicated the viruses found in the file, the fingerprint data would have included a flag indicating which data (viruses) was included within said fingerprint data."

Applicant respectfully disagrees and again asserts that applicant claims that "said resource data of said packed computer file is processed to generate fingerprint data" where "said generated fingerprint data includes a flag indicating which data is included within said generated fingerprint data," when read in context. Clearly, the fingerprint claimed by applicant is generated from processed resource data, in the context claimed, and is not merely representative of the file during comparison, as alleged by the Examiner. Furthermore, applicant respectfully asserts that the flags disclosed in Cozza are only included in the scan information cache file which "includes data that has been accumulated during the scanning of files" (see Col. 5, lines 17-19 in Cozza), and not in the file itself in which a virus was found. To this end, even a hash of the file that includes

the viruses would not itself include a flag indicating any sort of viruses, as argued by the Examiner. Thus, it would not have been obvious for "said generated fingerprint data [to include] a flag indicating which data is included within said generated fingerprint data," in the context claimed.

Still yet, with respect to the independent claims, the Examiner has relied on the following excerpt from the Cozza reference to make a prior art showing of applicant's claimed technique "wherein said generated fingerprint data includes a location within said resource data of said packed computer file of an entry specifying a program resource item having a largest size" (see this or similar, but not necessarily identical language in the independent claims).

> "If a file's cache information is not marked as having been previously infected by some virus which changes a file's resource fork size, then **the file's current resource fork size is compared with the resource fork size stored in the file's cache information in step 66 to see if they are within some predetermined tolerance.** The tolerance in this step is determined based upon the size of viruses infecting a file's resource fork on the Apple Macintosh computer, upon the type of file being infected, and upon the typical size changes that might occur in Macintosh applications and other executable files due to minor changes by which the file might modify itself. This tolerance may vary from one file to another depending on file type and other factors. If these sizes are not within the predetermined tolerance, then flags are set for all viruses that might cause this file's resource fork to change size when infecting it in step 68." (Cozza, Col. 6, lines 29-45 – emphasis added)

Applicant respectfully asserts that the excerpt from Cozza relied upon by the Examiner merely discloses comparing "the file's current resource fork size ... with the resource fork size stored in the file's cache information in step 66 to see if they are within some predetermined tolerance" (emphasis added). Clearly, simply disclosing checking a current or stored resource fork size, as in Cozza, fails to even suggest a technique "wherein said generated fingerprint data includes a **location** within said resource data of said packed computer file of an entry specifying a program resource item having a largest size" (emphasis added), as claimed by applicant.

Moreover, with respect to the independent claims, the Examiner has relied on Figure 5 from Cozza and Page 21, "PE File Base Relocations" in Pietrek to make a prior art showing of applicant's claimed technique "wherein said generated fingerprint data includes a checksum value calculated in dependence upon: a number of program resource items specified beneath each node within hierarchically arranged resource data of said packed computer file" (see this or similar, but not necessarily identical language in the independent claims).

Applicant respectfully asserts that the excerpt from Cozza relied upon by the Examiner merely discloses "cache data structures" where a "new cache includes data that has been accumulated during the scanning of files, data about the cache itself, i.e. its version, volume creation date, file id, and checksum, and scan information for each file scanned" (Column 5, lines 17-21 - emphasis added). Additionally, applicant notes that the excerpt from Pietrek relied on by the Examiner merely teaches that "[w]hen the linker creates an EXE file, it makes an assumption about where the file will be mapped into memory" and "[b]ased on this, the linker puts the real addresses of code and data items into the executable file" (Page 21, "PE File Base Relocations," paragraph 1).

However, merely disclosing a cache data structure which includes a cache checksum, in addition to teaching that a linker puts code and data into an executable file, fails to even *suggest* a technique "wherein said generated fingerprint data includes a checksum value calculated in dependence upon: a number of program resource items specified beneath each node within hierarchically arranged resource data of said packed computer file" (emphasis added), as claimed by applicant.

Additionally, the Examiner has argued that it "would have been obvious that the checksum of the file in this combination would have been dependent upon a number of said program resource items specified beneath each node within hierarchically arranged resource data of said packed computer file... as Pietrek teaches that Win32 PE files are arranged in such a manner, as seen [in] Pietrek Fig. 5 and Table 13."

Applicant respectfully disagrees and points out that Figure 5 and Table 13 from Pietrek merely show a resource directory hierarchy example and the resources hierarchy for CLOCK.EXE. However, the table and figure relied upon by the Examiner in no way suggest "**fingerprint data** [that] includes a checksum value calculated in dependence upon: a number of program resource items specified beneath each node within hierarchically arranged resource data of said packed computer file" (emphasis added), as claimed by applicant.

Applicant respectfully asserts that at least the first and third elements of the *prima facie* case of obviousness have not been met, since it would be *unobvious* to combine the references, as noted above, and the prior art reference excerpts, as relied upon by the Examiner, fail to teach or suggest all of the claim limitations, as noted above. Thus, a notice of allowance or a proper prior art showing of all of applicant's claim limitations, in combination with the remaining claim elements, is respectfully requested.

Applicant further notes that the prior art is also deficient with respect to the dependent claims. For example, with respect to Claim 12 et al., the Examiner has relied on the field VOLUMECRDATE in Figure 5 from the Cozza reference to make a prior art showing of applicant's claimed technique "wherein said generated fingerprint data includes timestamp data indicative of a time of compilation of said known computer program."

Applicant respectfully asserts that the figure from Cozza relied upon by the Examiner merely teaches a field VOLUMECRDATE that represents the "creation date of [a] volume on which [a] file resides," the field being part of the "Macintosh Scan Information Cache File Structure" (Fig. 5). However, merely disclosing a field describing the creation date of the **volume** in which a file resides, as in Cozza, fails to teach a technique "wherein said generated fingerprint data includes timestamp data indicative of a time of compilation of said **known computer program**" (emphasis added), as claimed by applicant. Further, applicant respectfully points out that in the Office Action dated 09/01/2006, the Examiner even admitted that "[t]he combination of

Cozza and Hypponen... failed to disclose providing a time of compilation in the fingerprint data" (emphasis added). Thus, Cozza simply does not meet applicant's specific claim language.

In addition, with respect to dependent Claim 14 et al., the Examiner has admitted that Cozza, Arnold, and Peitek do not disclose applicant's specific claim language, and thus has simply dismissed the same under Official Notice. Specifically, the Examiner has argued that "SHA, which shifts 1 bit to the left after each operation, was a well known checksum in the art at the time of the invention, and as such it would have been obvious to the ordinary person skilled in the art to have used SHA as the checksum."

Applicant respectfully disagrees and asserts that even assuming *arguendo* that it would have been obvious to shift 1 bit after each operation in calculating a checksum, such still does not address applicant's specific claim language, namely that "said checksum value is rotated between each item being added into said checksum" (emphasis added), as claimed. Thus, applicant's claimed technique simply would not have been obvious.

Applicant thus formally requests a specific showing of the subject matter in ALL of the claims in any future action. Note excerpt from MPEP below.

"If the applicant traverses such an [Official Notice] assertion the examiner should cite a reference in support of his or her position." See MPEP 2144.03.

Again, since at least the first and third elements of the *prima facie* case of obviousness have not been met, as noted above, a notice of allowance or specific prior art showing of each of the foregoing claim elements, in combination with the remaining claimed features, is respectfully requested.

To this end, all of the independent claims are deemed allowable. Moreover, the remaining dependent claims are further deemed allowable, in view of their dependence on such independent claims.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 505-5100. The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 50-1351 (Order No. NAI1P467).

Respectfully submitted,
Zilka-Kotab, PC

/KEVINZILKA/

Kevin J. Zilka
Registration No. 41,429

P.O. Box 721120
San Jose, CA 95172-1120
408-505-5100